

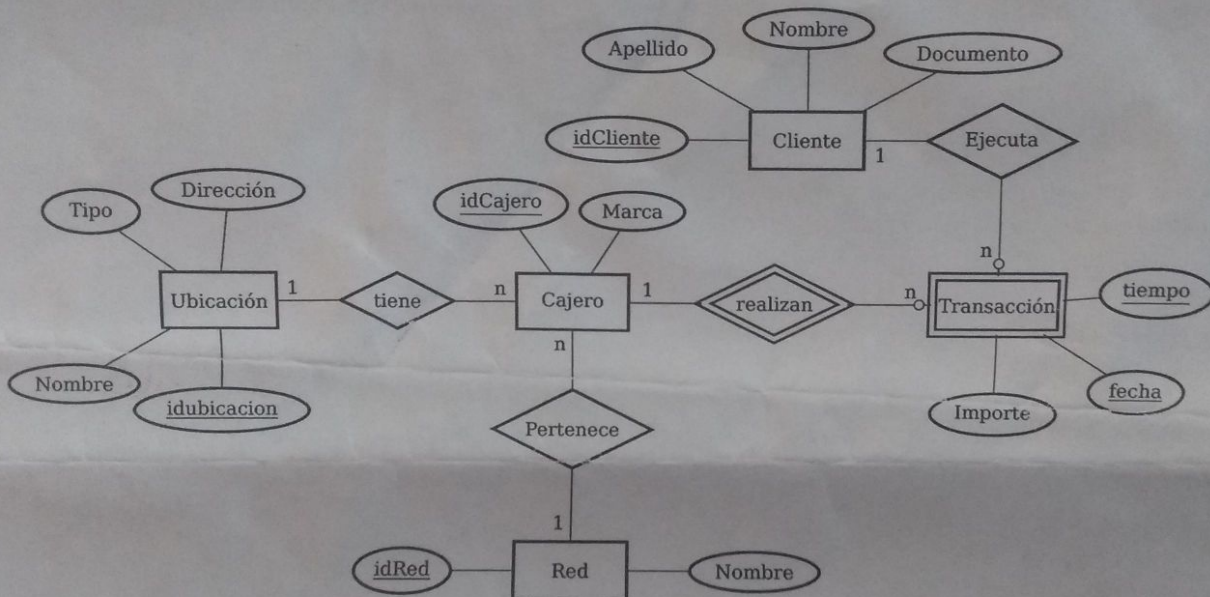
## 2 Parcial - Bases de Datos - 13/11/2019

- Debe identificarse **cada** hoja con nombre, apellido, LU y su **número de orden**.
- Complete la primera hoja con la cantidad total de hojas entregadas y numere todas las hojas.
- Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Para que un ejercicio sume puntos **no deben cometerse errores conceptuales graves**.
- La **interpretación** del enunciado forma parte de la evaluación.
- El parcial es a libro **cerrado**. Justifique sus respuestas.

**Criterio de Aprobación:** Se aprueba con 7. Ejercicio 1 5ptos, Ejercicio 2 5ptos,

### 1. NOSQL

Dado el siguiente DER que modela los datos históricos de cajeros automáticos y sus transacciones:



#### a) Document Database

(i) Dibujar el diagrama de interrelación de documentos, justificando las decisiones tomadas. Se debe tomar en cuenta que en general se debe buscar para una fecha dada las transacciones realizadas. Además es importante saber en forma eficiente para una ubicación, cuales son los cajeros que posee y la red a la que pertenece. Tener en cuenta que se trata de una base de datos histórica y el cajero nunca cambia de red.

(ii) Especificar en JSON Schema el tipo de documento Ubicación

#### b) Realizar el diseño Column Family haciendo el diagrama de Chebotko para las siguientes consultas:

- Las transacciones (todos los datos, incluido el cajero) para un cliente dado en un rango de fechas.
- Todas las ubicaciones donde una red dada tiene cajeros.

c) Se desea hacer un frontend web para acceder a la base de datos. Para ello se debe guardar los usuarios y passwords así como los datos de sesión (es decir fecha de ingreso y tablas accedidas). Diseñar una base de datos Key Value para sostener esta información .



## 2. Concurrency and Recoverability

a) Dada la siguiente historia sobre las transacciones  $T_1, T_2$  y  $T_3$

$H_1 = w_1(A); u_1(A); r_1(A); w_1(B); c_1; u_1(B); r_1(B); w_2(A); u_2(A); r_2(A); c_2; w_3(A); w_3(B); u_3(A); u_3(B); c_3$

(i) Construir el  $SG(H_1)$  (grafo de precedencia). Indicar si  $H_1$  es **SR** (serializable) y en caso afirmativo indicar las historias seriales equivalentes. JUSTIFIQUE

(ii) Indicar el nivel de recuperabilidad que tiene  $H_1$ : No Recuperable, Recuperable, ACA o Estricta. JUSTIFIQUE

(iii) ¿Cumple alguna de las Transacciones con algunos de los siguientes protocolos 2PL, 2PLE, 2PLR? JUSTIFIQUE

b) Suponga la siguiente historia tentativa parcial sobre un planificador con timestamp sin multiversion

$H_2 = r_1(A); c_1; r_3(C); r_2(C); w_3(A); r_2(B); r_3(B); w_2(B); c_3; w_2(A); c_2$

(i) Analice (JUSTIFIQUE) que pasaría si quisiéramos que orden serial equivalente sea

- $T_1, T_2, T_3$
- $T_1, T_3, T_2$

(ii) Para el punto anterior: ¿Cambiaría algo si el planificador fuera multiversion? JUSTIFIQUE

c) Suponga que, luego de que se produzca una falla en un sistema de bases de datos, se encuentran los siguientes registros en el log.

1. < START  $T_1$  >
2. <  $T_1, X, 41$  >
3. < START  $T_2$  >
4. <  $T_2, Z, 17$  >
5. <  $T_1, Y, 3$  >
6. < COMMIT  $T_2$  >
7. < START  $T_3$  >
8. <  $T_3, Z, 7$  >
9. < COMMIT  $T_1$  >
10. < START  $T_4$  >
11. <  $T_3, Y, 90$  >
12. <  $T_4, X, 6$  >

(i) Considerando el caso en que la estrategia de recuperación utilizada es UNDO-logging:

A) Describa las acciones que deberían realizarse para devolver la base de datos a un estado consistente. ¿Qué valores quedan almacenados en cada ítem de datos luego de finalizar el proceso?

B) Suponga que se utiliza checkpointing no-quietescente y se agrega la entrada: < START CKPT( $T_1, T_3$ ) > entre las líneas 7 y 8:

¿Podría aparecer la entrada < END CKPT > entre las líneas 10 y 11? ¿Por qué?

(ii) Ídem ítem anterior, pero considerando la estrategia REDO-logging.